I claim:

1. A method comprising:

dividing an executable software program in memory into an executable image, a data image, and an execution history image; and

classifying a first statement in said execution history image into one of a mutable statement and an immutable statement.

2. The method of claim 1 further comprising:

executing cryptographic integrity checks on said immutable statement; and

encrypting said immutable statement.

3. The method of claim 1 further comprising:

executing executable statements, local constants, and singly de-referenced pointers in said executable image;

processing data, data write-backs, and data read-backs in said data image, wherein said data image is accessed from said executable image using a computed offset into said data image from said executable image;

logging the usage of said first statement into said execution history image; and

terminating said executable software program when a mutable statement changes an immutable statement in memory.

4. The method of claim 3 further comprising re-mapping said first statement into a new executable software program wherein immutable statements are stored in locations in memory such that executing mutable statements cannot overwrite mutable statements.

5. The method of claim 1 wherein classifying further comprises mapping said first statement into one of an executable statement, a single data constant, a singly de-referenced pointer to data, an immutable multiply de-referenced pointer to data, an immutable data location, a mutable pointer location, a mutable data location, an input buffer, an output buffer, and an unused location.

6. A method comprising:

dividing an executable software program in memory into an executable image, a data image, and an execution history image;

executing executable statements, local constants, and singly de-referenced pointers in said executable image; and

processing data, data write-backs, and data read-backs in said data image, wherein said data image is accessed from said executable image using a computed offset into said data image from said executable image.

7. The method of claim 5 further comprising logging the usage of a first statement into said execution history image as said statement is processed.

8. An apparatus comprising:

a processor;

a memory connected to said processor;

an executable software program residing in said memory; and

an operating system residing in said memory and executing on said processor, wherein said operating system comprises a software module for:

dividing an executable software program in memory into an executable image, a data image, and an execution history image; and

classifying a first statement in said execution history image into one of a mutable statement and an immutable statement.

9. The apparatus of claim 8 wherein said operating system further comprises a software module for:

executing cryptographic integrity checks on said immutable statement; and

encrypting said immutable statement.

10. The apparatus of claim 8 wherein said operating system further comprises a software module for:

executing executable statements, local constant, and singly de-referenced pointers in said executable image;

processing data, data write-backs, and data read-backs in said data image, wherein said data image is accessed from said executable image using a computed offset into said data image from said executable image;

logging the usage of said first statement into said execution history image; and

terminating said executable software program when a mutable statement changes an immutable statement in memory.

11. The apparatus of claim 10 wherein said operating system further comprises a software module for re-mapping said first statement into a new executable software program wherein immutable statements are stored in locations in memory such that executing mutable statements cannot overwrite mutable statements.

12. The apparatus of claim 8 wherein classifying further comprises mapping said first statement into one of an executable statement, a single data constant, a singly de-referenced pointer to data, an immutable multiply de-referenced pointer to data, an immutable data location, a mutable pointer location, a mutable data location, an input buffer, an output buffer, and an unused location.

13. An apparatus comprising:

a processor;

a memory connected to said processor;

an executable software program residing in said memory; and

an operating system residing in said memory and executing on said processor, wherein said operating system comprises a software module for:

dividing an executable software program in memory into an executable image, a data image, and an execution history image; and

executing a statement in said executable image, wherein said executing further comprises executing data write-backs and data read-backs in said data image, and wherein said data image is accessed using a computed offset into said data image from said executable image.

14. The apparatus of claim 13 wherein said operating system further comprises a software module for logging the usage of said statement into said execution history image as said statement is executed from said executable image.

15. An apparatus comprising:

a host computer comprising a memory and a processor;

an executable software program residing in said memory; and

an operating system residing in said memory and executing on said processor, wherein said operating system comprises a software module for:

dividing an executable software program in memory into an executable image, a data image, and an execution history image; and

classifying a first statement in said execution history image into one of a mutable statement and an immutable statement.

16. The apparatus of claim 15 wherein said operating system further comprises a software module for:

executing cryptographic integrity checks on said immutable statement; and

encrypting said immutable statement.

17. The apparatus of claim 15 wherein said operating system further comprises a software module for:

executing executable statements, local constant, and singly de-referenced pointers in said executable image;

processing data, data write-backs, and data read-backs in said data image, wherein said data image is accessed from said executable image using a computed offset into said data image from said executable image;

logging usage of said first statement into said execution history image; and

terminating said executable software program when a mutable statement changes an immutable statement in memory.

18. The apparatus of claim 17 wherein said operating system further comprises a software module for re-mapping said first statement into a new executable software program wherein immutable statements are stored in locations in memory such that executing mutable statements cannot overwrite mutable statements.

19. The apparatus of claim 15 wherein classifying further comprises mapping said first statement into one of an executable statement, a single data constant, a singly de-referenced pointer to data, an immutable multiply de-referenced pointer to data, an immutable data location, a mutable pointer location, a mutable data location, an input buffer, an output buffer, and an unused location.

20. An apparatus comprising:

a host computer comprising a memory and a processor;

an executable software program residing in said memory; and

an operating system residing in said memory and executing on said processor, wherein said operating system comprises a software module for:

dividing an executable software program in memory into an executable image, a data image, and an execution history image; and

executing a statement in said executable image, wherein said executing further comprises executing data write-backs and data read-backs in said data image, and wherein said data image is accessed using a computed offset into said data image from said executable image.

21. The apparatus of claim 20 wherein said operating system further comprises a software module for logging the usage of said statement into said execution history image as said statement is executed from said executable image.

22. A machine-readable medium comprising a software module for:

dividing an executable software program in memory into an executable image, a data image, and an execution history image; and

classifying a first statement in said execution history image into one of a mutable statement and an immutable statement.

23. The machine-readable medium of claim 22 further comprising a software module for:

executing cryptographic integrity checks on said immutable statement; and

encrypting said immutable statement.

24. The machine-readable medium of claim 22 further comprising a software module for:

executing executable statements, local constant, and singly de-referenced pointers in said executable image;

processing data, data write-backs, and data read-backs in said data image, wherein said data image is accessed from said executable image using a computed offset into said data image from said executable image;

logging the usage of said first statement into said execution history image; and

terminating said executable software program when a mutable statement changes an immutable statement in memory.

25. The machine-readable medium of claim 24 further comprising a software module for re-mapping said first statement into a new executable software program wherein immutable statements are stored in locations in memory such that executing mutable statements cannot overwrite mutable statements.

26. The machine-readable medium of claim 22 wherein classifying further comprises mapping said first statement into one of an executable statement, a single data constant, a singly de-referenced pointer to data, an immutable multiply de-referenced pointer to data, an immutable data location, a mutable pointer location, a mutable data location, an input buffer, an output buffer, and an unused location.

27. A machine-readable medium comprising a software module for:

dividing an executable software program in memory into an executable image, a data image, and an execution history image; and

executing a statement in said executable image, wherein said executing further comprises executing data write-backs and data read-backs in said data image, and wherein said data image is accessed using a computed offset into said data image from said executable image.

28. The machine-readable medium of claim 27 further comprising a software module for logging the usage of said statement into said execution history image as said statement is executed from said executable image.